REMARKS

In view of th following remarks, Applicant r spectfully requests reconsideration and allowance of the subject application. This reply is believed to be fully responsive to all issues raised in the Office Action mailed March 4, 2004.

Claim Rejections

Rejections Under 35 U.S.C. §103

Claims 1-20 were rejected under 35 U.S.C. §103(a) as being unpatentable over the U.S. Patent No. 5,944,838 to Jantz ("Jantz"). Applicants respectfully traverse these rejections.

Jantz cannot render obvious claim 1 because, contrary to the assertion in the Action, Jantz neither discloses nor suggests the subject matter recited in claim 1. Claim 1 recites the following limitations:

> 1. A method of controlling a failover process in a data storage system including a host, a host bus adapter, a communication fabric including data paths, and standby and active storage controllers, comprising:
> detecting with the host bus adapter a failover condition;
> responsive to the detecting, operating the host bus adapter to match the failover condition to a particular failover action in a failover rule set; and
> performing with the host bus adapter the matched failover action.

The Action acknowledges that Jantz fails to disclose or suggest detecting a failover condition with a host bus adapter. However, the Action asserts that it would have been obvious to one of skill in the art to modify Jantz to detect and respond to a failover condition with a host bus adapter,

ostensibly "to encompass control of a large number of I/O path elements in its failure recovery t chniques." Applicants disagree.

Conventionally, failover processing has been performed in control software operating as part of a host computer's operating system, or as middleware positioned between application programs that consume storage and lower-layer control software (or firmware) for operating a storage system. Consistent with this convention, Jantz explicitly teaches, at column 8, lines 29-45, that failure *detection* is performed by the low-level device driver 310, and that failure *processing* is performed within the host computer's operating system:

> Low level device driver 310 of FIG. 3 is asynchronously operable to process the I/O request buffered (queued) in its path A dispatch queue 306. Each I/O request in the dispatch queue is processed in sequence to perform the I/O operation identified therein. Low level device driver 310 returns a status message to RDAC 304 indicating the processed I/O request has either succeeded or failed. In case of a failure, low level device driver 310 has performed any required retry of operations to assure that the operation cannot be successfully performed. The processing of low level device driver 310 is typically provided by interface functions within the operating system of the host computer (host system API). Such functions are typically standardized and in compliance with one or more industry standards for such functionality (e.g., UNIX/POSIX, MS Windows.RTM., etc.). The operation of and interface to low level device driver 310 is therefore well known to those skilled in the art.

As an extension of the host computer's operating system, device driver 310 has the visibility to detect the failure of any I/O path element visible to the host computer. Contrary to the rationale asserted in the Action, detecting failover conditions in the host bus adapter actually reduces the number of I/O path elements for which failover conditions may be detected.

For at l ast this reason, on skilled in the art would not b motivated to modify Jantz as sugg sted in the Action.

Claim 1 further recites the limitations of *responsive to the detecting, operating the host bus adapter to match the failover condition to a particular failover action in a failover rule set; and performing with the host bus adapter the matched failover action.* The Action asserts that Jantz discloses this subject matter, and cites column 8, lines 45-67 and column 9, lines 10-67. Applicants disagree.

Nothing in the cited text discloses *matching the failover condition to a particular failover action in a failover rule set.* Indeed, Jantz neither discloses nor suggests detecting specific failover conditions; rather, Jantz merely detects whether a failure has occurred. In addition, Jantz always implements the same failover procedure (which is illustrated in Fig. 5) in response to a failure. Therefore, there can be no motivation in Jantz to match a failover condition to a particular failover action in a failover rule set. Furthermore, Applicants note that claim 1 explicitly recites performing these operations *in the host bus adapter.* Jantz neither discloses nor suggests performing failover operations in the host bus adapter, and the Action provides no rationale to indicate why one of skill in the art would modify Jantz to perform these operations in the host bus adapter.

For at least these reasons the rejection of claim 1 is improper and should be withdrawn.

Applicant traverses the rejection of claim 2. Claim 2 recites the limitation that *the detecting, operating, and the failover action performing are*

compl ted without acts initiated by the host. The Action asserts that Jantz t aches this limitation, and cites column 1, lines 25-48. Applicants disagree. The cited text reads as follows:

> Redundant I/O paths can take any of a number of forms, including but not limited to SCSI buses, host adapters, or RAID controllers. In a system with redundant I/O paths connecting a storage controller to the storage device(s), there is a control sub-subsystem which manages the redundant paths referred to herein "Redundant Dual-Active Control" (RDAC). An RDAC control subsystem is often a layer of software in a hierarchical layering of control software which provides the interface between the host systems and storage subsystems.
>
> One skilled in the art will recognize that the RDAC layer is a logical component, typically embodied as a software module. The RDAC layer typically operates within either the host system (as part of the operating system) or may be operable within intelligent I/O adapters in the host as well as embedded storage controllers within the storage subsystem. The physical components on which the RDAC layer is operable are not particularly relevant to the layered architecture of which the RDAC layer is a component. It is generally desirable that the RDAC layer operate at a higher level thus enabling it to encompass control of a larger number of I/O path elements in its failure recovery techniques.

Nothing in this text discloses or suggests that *the detecting, operating, and the failover action performing are completed without acts initiated by the host.* Further, contrary to the assertion in the Action, Jantz explicitly teaches, at column 8, lines 29-45 (excerpted above) that failure detection and response are all initiated in the host computer's operating system or on associated device drivers. Therefore, the rejection of claim 2 is improper and should be withdrawn.

Applicant traverses the rejection of claim 3. Claim 3 recites the limitation that *the detecting includes identifying a particular failure type and wherein the particular failover action is selected from an action subset*

corresponding to the p rticular failure type. Th Action ass rts that Jantz t aches this limitation, and cites column 1, lines 25-48, and column 2, lines 19-33. Applicants disagree. Column 1, lines 25-48 are excerpted above. Column 2, lines 19-33 reads as follows:

> A variety of failures could occur such that the RDAC layer might not be able to access the storage device via one of the redundant I/O paths (e.g., via the preferred I/O path). A software failure in the low level disk driver module is one example of such a failure. Or for example a hardware failure might occur in the physical connection from the disk driver module to the disk array. In general, all such failures which render the I/O path unusable to the RDAC layer will be identified herein as I/O path failures. An I/O path which has failed is also referred to herein as a bad path or failed I/O path while an operational I/O path is also referred to herein as a good path or operational I/O path. In general, when the RDAC layer becomes aware of such a failure in an I/O path (the bad path), failed I/O requests are redirected (retried) on the other I/O path (the good path).

Nothing in this text discloses or suggests that *the detecting includes identifying a particular failure type and wherein the particular failover action is selected from an action subset corresponding to the particular failure type.* Further, as noted above, Jantz always implements the same failover procedure (which is illustrated in Fig. 5) in response to a failure. Therefore, there can be no motivation in Jantz to identify a particular failure type or to select a failover action from an action subset corresponding to a particular failure type. For at least these reasons the rejection of claim 3 is improper and should be withdrawn.

Applicant traverses the rejection of claim 4. Claim 4 recites the limitation that *the failure type is selected from the group consisting of inter-controller link down, the active storage controller failed, the standby*

controller failed, an active path failed, and a standby path failed. Th Action asserts that Jantz t aches this limitation, and cit s column 2, lines 19-33 (excerpted above). Applicants disagree. Nothing in column 2, lines 19-33 discloses or suggests that *the failure type is selected from the group consisting of inter-controller link down, the active storage controller failed, the standby controller failed, an active path failed, and a standby path failed.* Therefore, the rejection of claim 4 is improper and should be withdrawn.

Applicants traverse the rejection of claim 5. Claim 5 recites the limitation that *prior to the performing, determining with the host bus adapter if all active paths have failed and if all active paths determined failed, skipping the failover action performing when the host bus adapter determines either all other available paths have failed or a standby path is marked as unusable.* The Action acknowledges that Jantz is silent as to these limitations, but asserts these limitations are inherent in Jantz.

Applicants agree that Jantz neither discloses nor suggests the limitations recited in claim 5, but Applicants specifically traverse the assertion that these limitations are inherent in Jantz. Inherency is governed by MPEP 2112, which provides as follows:

> The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. *In re Rijckaert*, 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993) . . . *In re Oelrich*, 666 F.2d 578, 581-82, 212 USPQ 323, 326 (CCPA 1981). "To establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.' " *In re*

> *Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999)
>
> "In relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art." Ex parte Levy, 17 USPQ2d 1461, 1464 (Bd. Pat. App. & Inter. 1990) (emphasis in original) . . .

Applicants submit that the evidence of record is insufficient to establish obviousness by inherency. The assertions in the Action regarding the operation of Jantz are pure conjecture and are unsupported by any facts.

Applicant traverses the rejection of claim 6. Claim 6 recites the limitation that *after the failover action performing, operating the host bus adapter to initiate failback when a controller in a preferred slot is replaced, when the controller in the preferred slot is rebooted, and when unusable paths become usable.* The Action asserts that Jantz teaches this limitation, and cites column 7, lines 5-24. Applicants disagree. The cited text reads as follows:

> RDAC 304 controller preferably maintains a single pending I/O queue 312 in which a copy of each I/O request is maintained until it is completed. Any particular I/O request in the pending I/O queue 312 could be pending on path A or on path B depending on which I/O path was selected for initiation of the request. Alternatively, RDAC 304 could maintain a separate pending I/O queue for path B as well and thus it might undertake the same process for I/O requests directed to path B (whether originally directed thereto or redirected thereto in response to failover restart). One skilled in the art will recognize that the principle of maintaining a pending I/O queue associated with an I/O path may be extended to any number of alternate redundant I/O paths. Further, pending I/O queue 312 may be implemented by any of several techniques well known to those skilled in the arts. Various well known software data structures and algorithms and hardware structures can be utilized in creating such a queue, including for example: a

linked list, a priority queue, hardware FIFO circuits, data tables, and many others.

Nothing in this text discloses or suggests *after the failover action performing, operating the host bus adapter to initiate failback when a controller in a preferred slot is replaced, when the controller in the preferred slot is rebooted, and when unusable paths become usable.* Therefore, the rejection of claim 6 is improper and should be withdrawn.

Applicant traverses the rejection of claim 7. Claim 7 recites the limitation of *performing load distribution with the host bus adapter between the host and the controllers.* The Action asserts that Jantz teaches this limitation, and cites column 5, lines 28-41. Applicants disagree. The cited text reads as follows:

> As shown in FIG. 1, bus 105, control module 110, and bus 115 form a first I/O path between host system 104 and disk array 116. Bus 106, control module 112, and bus 114 form a second (redundant) I/O path between host system 104 and disk array 116. One of ordinary skill will further note that I/O adapters within host system 104, a first attached to bus 105 and a second attached to bus 106, may form yet another component of each of the redundant I/O paths. Further, it will be recognized that any number of I/O paths may connect host system 104 to disk array 116. FIG. 1 is therefore intended only as exemplary of one computing environment in which the methods of the present invention may be advantageously applied. Many similar computing environments will be recognized by those skilled in the art.

Nothing in this text discloses or suggests performing load balancing, much less *performing load distribution with the host bus adapter.* Therefore, the rejection of claim 7 is improper and should be withdrawn.

Applicants traverse the rejection of claims 8 and 19. The Action acknowledges that Jantz neither discloses nor suggests anti-thrashing rules

that prevent operations from being performed more than a preset number of times per monitoring interval. N v rtheless, the Action ass rts that thes limitations would have been obvious to one skilled in the art. Applicants disagree. Nothing in Jantz, or in the art in general, would motivate one skilled in the art to implement anti-thrashing rules in a host bus adapter, as recited in claims 8 and 19. Therefore, the rejections of claims 8 and 19 are improper and should be withdrawn.

Applicants traverse the rejection of claim 9. Jantz cannot render obvious claim 9 because, contrary to the assertion in the Action, Jantz neither discloses nor suggests the subject matter recited in claim 9. Claim 9 recites the following limitations:

> 9. A host bus adapter for managing failover and failback processes within a data storage system having a host server, a communication fabric, at least one active storage controller, and at least one standby storage controller, comprising:
>
> a connector linking the host bus adapter to a processor of the host server;
>
> a port linking the host bus adapter to the communication fabric configured for transmitting and receiving digital information; and
>
> a failover mechanism detecting a redundancy failure in the data storage system and in response, initiating failover actions.

Initially, Applicants note that Jantz neither discloses nor suggests *a connector linking the host bus adapter to a processor of the host server*, as recited in claim 9. The Action appears to assert that element 304 teaches this limitation. However, a close inspection of Jantz reveals that element 304 corresponds to a software module referred to as the RDAC, which is not *a connector linking the host bus adapter to a processor of the host server*, as recited in claim 9.

Similarly, Applicants note that Jantz neither discloses nor suggests *a port linking the host bus adapter to the communication fabric configured for transmitting and receiving digital information*, as recited in claim 9. The Action appears to assert that elements 302, 304, and 312 teach this limitation. However, a close inspection of Jantz reveals that element 302 corresponds to application software executing on a host computer, element 304 corresponds to a software module referred to as the RDAC, and element 312 corresponds to an I/O queue, none of which are *a port linking the host bus adapter to the communication fabric configured for transmitting and receiving digital information*, as recited in claim 9.

The Action further reiterates the obviousness arguments applied to claim 1 to support the rejection of claim 9. For the same reasons set forth with regard to claim 1, Applicants submit that Jantz cannot render obvious claim 9. For at least these reasons, Applicants submit that the rejection of claim 9 is improper and should be withdrawn.

Applicants traverse the rejections of claims 10, 11, and 12. As noted above, Jantz neither discloses nor suggests selective failover mechanisms, much less the specific limitations recited in claims 10, 11, and 12. To the contrary, Jantz always performs the same failover routine without regard to the type of failure. Therefore, these rejections are improper and should be withdrawn.

Applicant traverses the rejection of claim 13. Claim 13 recites the limitation that *the failover mechanism presents a single logical unit number (LUN) entity to operating system device drivers in the host processor that is*

discoverable a plurality of times and wherein the failover actions are initiated without prior communication with the host processor. Th Action ass rts that Jantz teaches this limitation, and cites column 5, lines 42-52 and column 6, lines 26-38, and column 1, lines 25-48 (excerpted above). Applicants disagree. The cited text reads as follows:

> FIG. 2 is a simplified block diagram depicting the flow of I/O requests in an RDAC system as known in the prior art. The application software 202 sends I/O requests to the RDAC 204. RDAC 204 then transfers the requests to low level disk driver 210 for further processing on a particular I/O path. Low level disk driver 210 then queues these requests in path A dispatch queue 206 for asynchronous processing by the low level disk driver 210. The low level disk driver 210 in turn controls the operation of the storage array (e.g., RAID LUNs not pictured) to process the I/O requests.
>
> As noted above, path A dispatch queue 206 (as well as path B dispatch queue 208) are constructs created and maintained within low level disk driver 210. The dispatch queues are used to buffer the high speed generation of I/O requests by the higher level software layers (e.g., application layer 202 and RDAC layer 204). Performance of low level disk driver 210 is gated by the relatively slow performance of the storage array (e.g., RAID LUNs not shown). The dispatch queues therefore serve to buffer I/O requests until low level disk driver 210 is ready to process the next request. The path A dispatch queue 206 may therefore have thousands of I/O requests waiting therein for processing by low level disk driver 210.

Nothing in this text discloses or suggests LUN presentation or management, much less presenting performing load balancing, much less *present[ing] a single logical unit number (LUN) entity to operating system device drivers in the host processor that is discoverable a plurality of times and wherein the failover actions are initiated without prior communication with the host processor.* Therefore, the rejection of claim 13 is improper and should be withdrawn.

Applicants traverse th rej ction of claim 14. Jantz cannot rend r obvious claim 14 b cause, contrary to the assertion in the Action, Jantz neither discloses nor suggests the subject matter recited in claim 14. Claim 14 recites the following limitations:

> 14. A data storage system with redundant data storage, comprising:
>
> a host computer device with a processor running operating system device drivers;
>
> a communication fabric for carrying digital data signals;
>
> an active controller controlling access by the host computer device to data storage devices; a standby controller controlling access by the host computer device to the data storage devices; and
>
> a host bus adapter linked to the host processor and the communication fabric for selecting a path through the communication fabric to one of the active and standby controllers for providing the operating system device drivers with access to the data storage devices, wherein host bus adapter is configured to initiate a failover action selected from a set of failover actions.

The Action acknowledges that Jantz fails to disclose or suggest detecting a failover condition with a host bus adapter. However, for the same reasons applied to claim 1, the Action asserts that it would have been obvious to one of skill in the art to modify Jantz to detect and respond to a failover condition with a host bus adapter. Applicants disagree for the same reasons set forth above with respect to claim 1.

In addition, claim 14 recites the limitation that the host bus adapter is *configured to initiate a failover action selected from a set of failover actions.* This limitation is neither disclosed nor suggested by Jantz. As noted above, Jantz always performs the same failover routine without regard to the type of failure. For at least these reasons, these rejections are improper and should be withdrawn.

Claims 15-20 were rej cted based on arguments similar to those applied to claims 2-8 and 10-13. Applicants traverse these rejections for the same reasons described above with respect to those claims.

CONCLUSION

Claims 1-20 are in believed to be in condition for allowance. Applicant respectfully requests reconsideration and prompt allowance of the present application. Should any issue remain that prevents immediate allowance of the application, the Examiner is encouraged to contact the undersigned attorney to discuss the unresolved issue.

Respectfully Submitted,

Dated: 5/10/04

Paul Mitchell
Lee & Hayes, LLP
Reg. No. 44,453
(509) 324-9256 x 237

Direct correspondence to:
Hewlett-Packard Company
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400